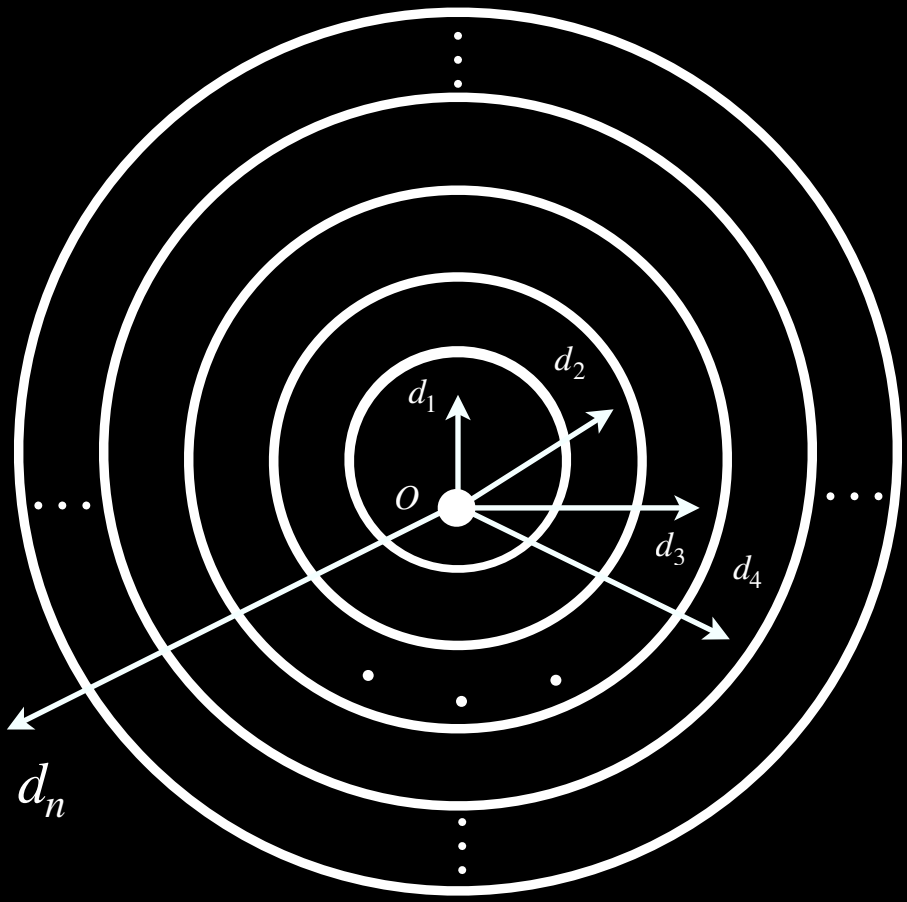
P = NP

MATHEMATICALLY EXPLAINED



INDIA SOALE

P = NP

MATHEMATICALLY EXPLAINED

INDIA SOALE

Copyright © India Soale, 2023

All illustrations, designs and theories outlined in the chapters are the work of India Soale. This ebook and all pictures contained therein are copyright material and must not be copied, transferred, reproduced, distributed, leased, licensed or used in any way other than for those which constitute strictly educational purposes that give credit to the author.

India Soale asserts the right to be identified as the author of this work in accordance with Copyright and Patent Law.
Under the terms of applicable copyright law, any unauthorised distribution or use of this text would be an infringement of the author's rights and would be liable in law accordingly.

www.indiasoale.com

CONTENTS

INTRODUCTION

The **P** versus **NP** Problem is a significant open problem in Computer Science and Decision Mathematics. It poses the question as to whether all problems, including those in **NP** are also in the **P**, i.e:

"whether every language accepted by some nondeterministic algorithm in polynomial time is also accepted by some deterministic algorithm in polynomial time" [Stephen Cook - "Statement of the Problem"]. The **NP** class is a set of problems which are solvable in nondeterministic polynomial time or verifiable in deterministic polynomial time. The **P** Class of problems, on the other hand, contains all decision problems which are solvable in deterministic polynomial time.

P is a set of problems which can be solved by an algorithm bounded by a certain polynomial in the length of the input.

All of the elements of **P** are languages such that:

$$\mathbf{P} = \{L \mid L = L(M) \text{ for a Turing machine '}M\text{' which runs in polynomial time}\}$$

A Turing machine is defined as a mathematical model that can implement any computer algorithm. The worst case running time for a computation 'M' to halt is denoted '$T_M(n)$' such that:

$$T_M(n) = \max\{t_M(w) \mid w \in \sum^n\}$$

$$w = \textit{input string}$$
$$M = \textit{Turing machine}$$
$$t_M(w) = \textit{number of steps for a computation 'M' to halt}$$

$$\sum = \textit{an alphabet}$$

We say that $\sum^n$ = set of strings of length '$\boldsymbol{n}$', such that $n \in \mathbf{N}$, where $\mathbf{N} = \{1, 2, 3, \ldots\}$. We say that '$M$' runs in polynomial time if there exists a non-negative 'p', such that for all 'n', $T_M(n)$ is bounded by a polynomial in 'n':

$$T_M(n) \in O(n^p)$$

For **P**, we say that $L(M)$, the language accepted by 'M' with associated alphabet "$\sum$" has the following definition:

$$L(M) = \{\boldsymbol{w} \in \sum^* : M \text{ accepts } w\}$$

In this Mathematical paper, I will demonstrate that all problems are solvable in Polynomial time and, as such, they all belong in **P**.

NP-complete problems are problems for which, up until now, no efficient algorithm has been found. One of the most famous examples is the 'Travelling Salesman Decision Problem' - a problem such, that given a graph with 'n' vertices, a salesman must visit all vertices and return back to the starting vertex in a length less than a given length, 'L.'

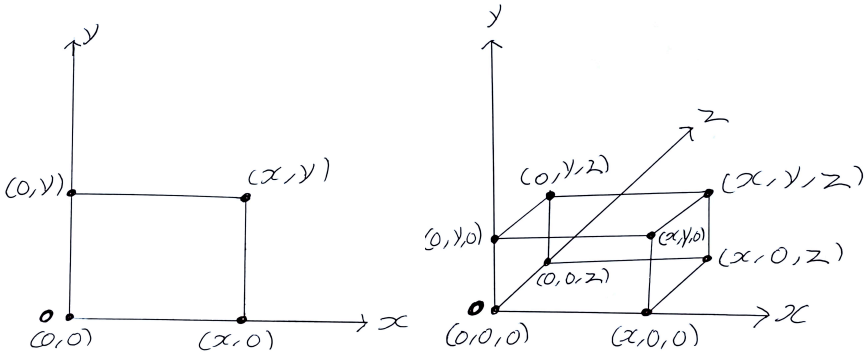
NP-hard problems, on the other hand, are problems which are hard to verify in polynomial time.

In this paper, I will show that problems in general can be solved by a polynomial-time algorithm called an 'Algorithm of Outcomes', i.e, an algorithm which can solve all problems which accept any input string 'w' of length 'n' in a maximum number of steps, a polynomial of 'n'. Though it may be tempting to skip ahead to Chapters 4 and 5, which all show the algorithm in great detail, I strongly recommend reading from the beginning, as the foundations are laid in Chapters 1 to 3.

1 — N-SPACES

ALL SPACES CAN BE MODELLED AS N-DIMENSIONAL

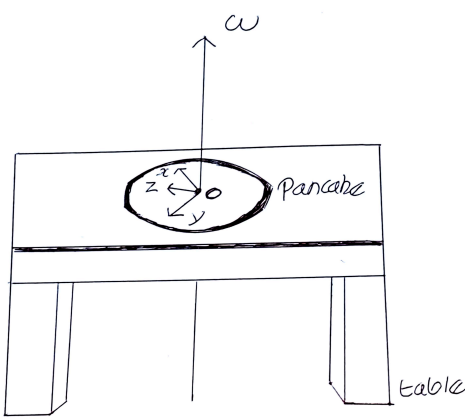
To understand the concept of an *n*-dimensional space on the ***n*** axes we must begin conceptualising dimensions beyond those most will be familiar with. The significance of n-dimensional spaces with the **P** versus **NP** problem will become clear later on. Mathematicians are familiar with the positive ***x,y*** axes on the left and also the positive ***z***-axis on the right.

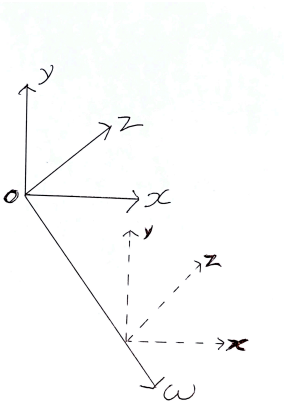


The key thing to notice about the ***z***-axis is that it is a shift of the ***x,y*** axes in ***z***. By treating the ***x,y*** axis as flat, we conceptualise the ***z***-axis. The same is true for the ***y***-axis, which we can conceptualise by treating the ***x***-axis as flat.

The same perspective can be used to demonstrate a fourth axis as to represent a fourth variable, and a fifth, and a sixth, and so on.

To conceptualise the fourth dimension '**w**', we must treat the ***x,y,z*** axes as pancake-flat. Picture the ***x,y,z*** axes as a pancake on a table and the ***w***-axis striking up through the centre of the pancake, along with the table:



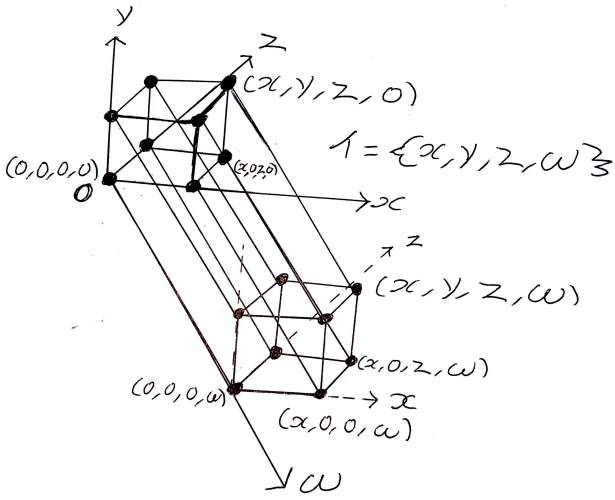


We can draw parallel dotted lines of the ***x,y,z*** axes along the **w**-axis to avoid confusing different axes.

To conceptualise this further, let us draw a four-dimensional space on the ***x,y,z.*****w** axes such as a tesseract.

First draw a cube in the ***x,y,z*** axes, draw the same cube in the ***x,y,z*** axes but a distance of '**w**' away from the ***x,y,z*** axes. Finally, we join the edges to complete the 4D cube.

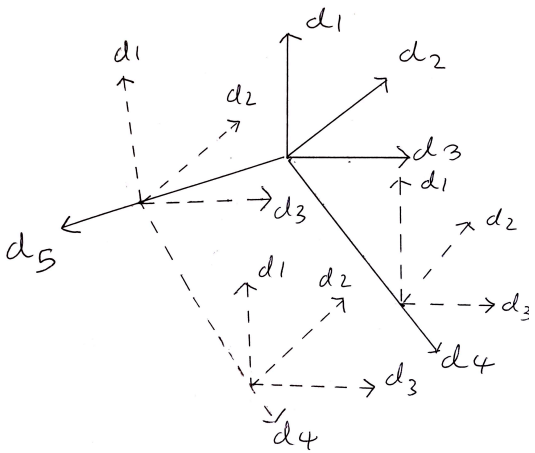
On the following page the four-dimensional space of the tesseract is illustrated.



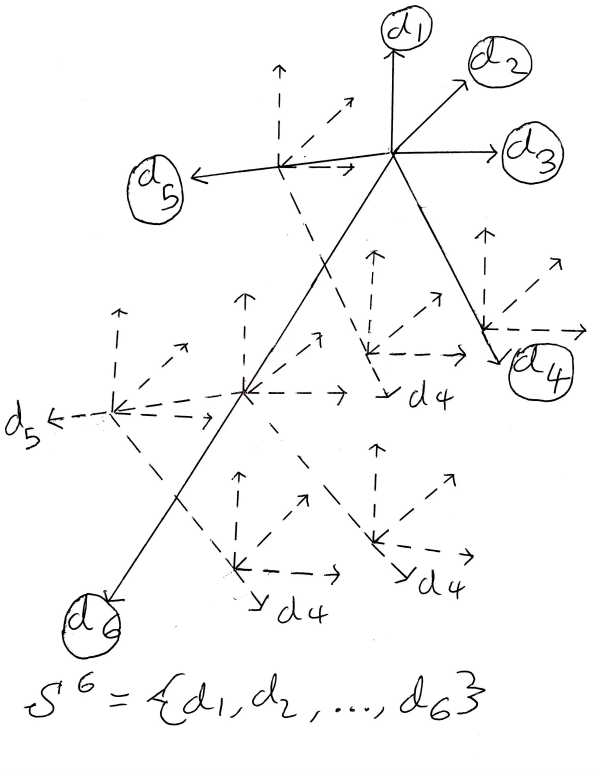
Let us denote '**x**' as 'd_1', '**y**' as 'd_2', '**z**' as 'd_3' and '**w**' as 'd_4.' We now have the set of four dimensions 'S^4':

$$S^4 = \{d_1, d_2, d_3, d_4\}$$

For the fifth dimension 'd_5', we do exactly the same as for 'd_3' and 'd_4.' By imagining the 'd_1,d_2,d_3,d_4' axes as a flat, we conceptualise the fifth dimension:



To show 'd_6', we model the 'd_1,d_2,…,d_5' axes as flat:

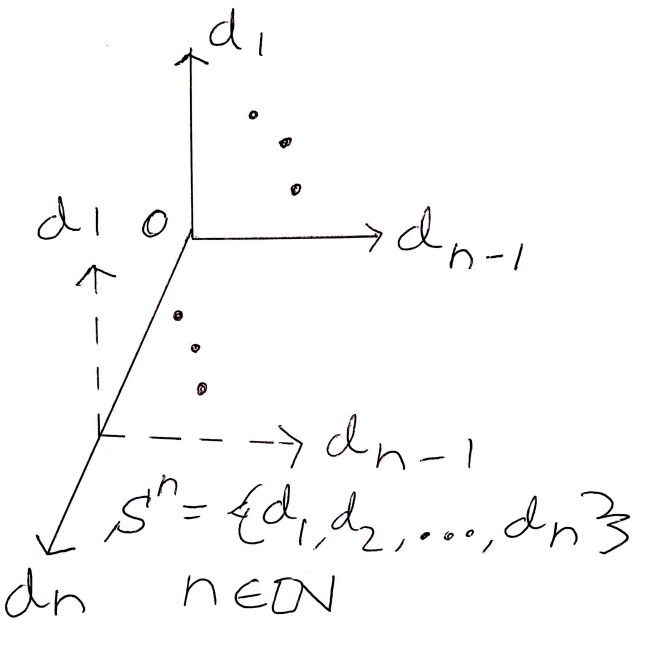


In the third chapter "Pancake Theory of Dimensions", I will demonstrate a compact way to represent dimensions.

We now shall denote all dimensions from d_1 to d_n as S^n with order 'n':

$$S^n = \{d_1, d_2, \ldots, d_n\},\ |S^n| = n \in \mathbf{N} = \{1, 2, \ldots\}$$

To draw n axes, we must therefore show 'd_n', by modelling the 'd_1,d_2,…,d_{n-1}' axes as flat, where the vertical dots on the axes represents the axes between 'd_1' and 'd_{n-1}':

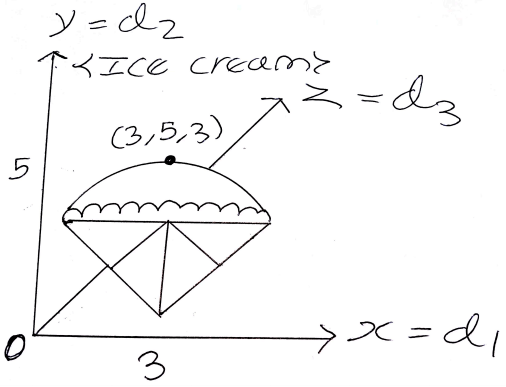


On seeing n-dimensional axes, we have demonstrated the existence of n axes, i.e, axes with values on each and every axis from d_1 to d_n.

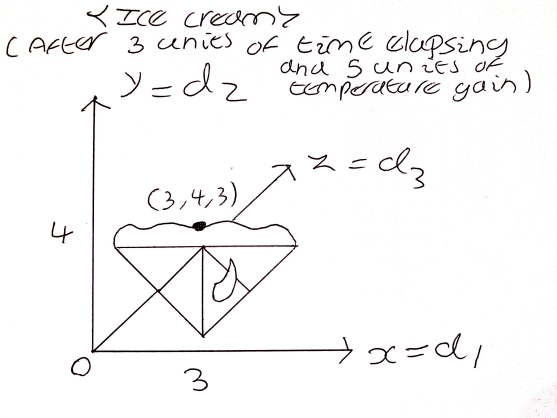
There is an endless potential for real-world applications of dimensions beyond the traditional two or three axes we are used to using.

Suppose we want to show the effects of an ice-cream cone melting after a certain length of time has elapsed. We can use d_1, d_2 and d_3 to represent the image of the ice cream, d_5 to represent 'temperature gain' and d_4 to represent 'time elapsed.' Just as a flat (two-dimensional) image of the ice cream changes as depth is added, the flat image of the ice cream also changes in d_1, d_2, d_3 as the temperature rises.

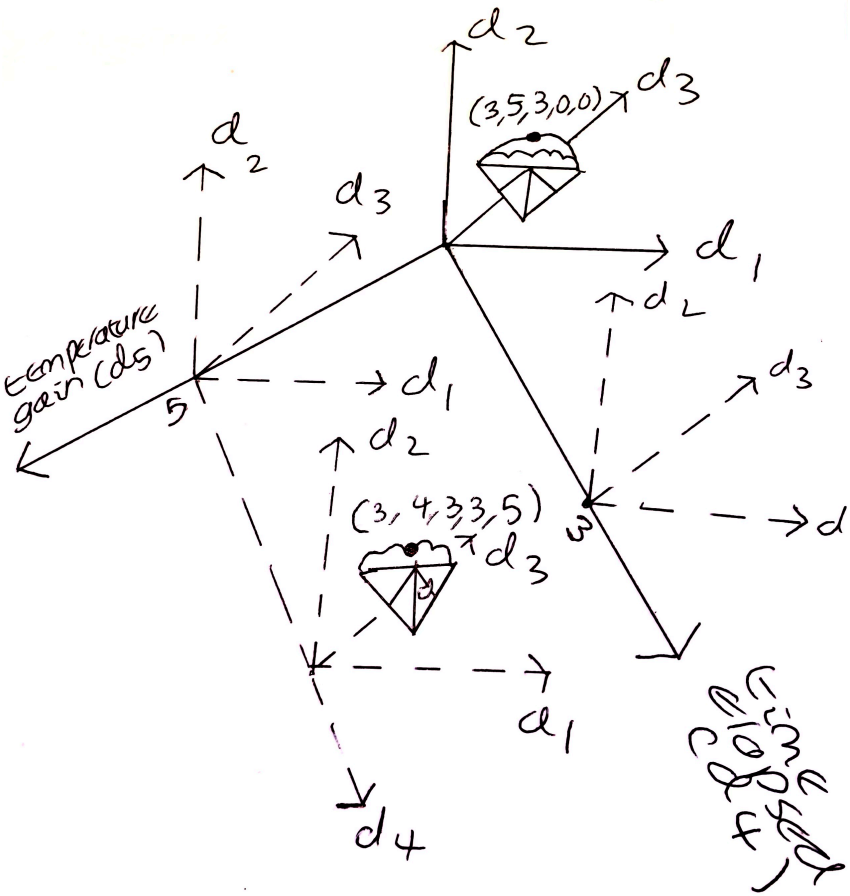
As the temperature rises, time elapses. If we suppose d_4, 'time elapsing' does not have a direct effect on the image of the space, the space can still be modelled to have a value in this axis. We could start by drawing the ice cream cone on the d_1,d_2,d_3 axes at the point that the 'time elapsed' and that the 'temperature gain' variables are at zero:



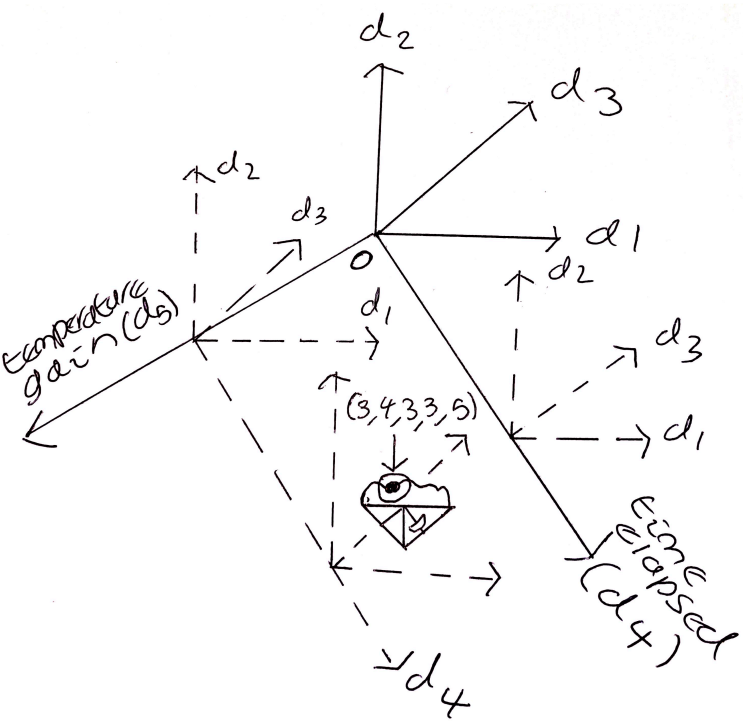
Next we can draw the ice cream cone on the d_1,d_2,d_3 axes after 3 units of 'time elapsed' and 5 units of 'temperature gain' have occurred. We now have the change in space of the ice cream after the occurrence of these two variables:



We can show both of these moments on one set of axes, by drawing both stages of the space to begin with and labelling the new top point of the ice cream (3, 4, 3, 3, 5).



Finally, we can remove the original space before the two variables (dimensions) were added:



We now have a real-world application of a five-dimensional space which concerns five different variables. We can denote the 5D space of the ice cream 'I^5' where:

$$I^5 = \{d_1, d_2, d_3, d_4, d_5\} =$$

{Width, Height, Depth, Time elapsed, Temperature gain}

I^5 is a space or function with order 5, with dimensions which have a range of possible values.

Let 'J' be a function of points on the ice cream after 3 units of time have passed and the temperature has gone up by 5 units.

Then we let the original 3D image of the ice cream on the d_1,d_2,d_3 axes be a function of 3 variables 'g(d_1, d_2, d_3)', then:

$$I^5 = \boldsymbol{J} = \text{f}(d_1, d_2, d_3, d_4 = 3, d_5 = 5) = \text{all points of an ice cream g}(d_1, d_2, d_3) \text{ when } d_4 = 3 \text{ and } d_5 = 5$$

Just as the dimensions of width, height and depth make up how a space appears in the real world, so does the temperature variable.

Reality, however, is far more complex. 'Time elapsed' and 'Temperature gained' are not the only variables that exist which may or may not have a direct effect on the space of the ice cream cone.

There are countless other variables we can think of, of which the ice cream cone has values, but which may or may not have an effect on the values of other dimensions which we have ignored, even if they are zero values or very trivial variables. For instance the *number of people that saw the ice cream*, or *cars that drove by as the ice cream was consumed* may not have any effect on the image of the ice cream space, but the ice cream space can be modelled to have parameters in these variables (dimensions), even if they are zero values.

There appears to be no limit on the variables of which a space can have parameters, regardless of how trivial or non-trivial the variables concerning the space may be. We propose that the ice cream cone has values in n variables, making it an n-dimensional space or n-space.

We model spaces to have the number of dimensions as those we have selected, 'n', which we can infinitely increase as we identify newer parameters, but can never reach infinity itself.

This will become clearer later on in the third chapter on the "Pancake Theory of Dimensions."

We can say the same for the largest known space, U^n, the Universe. If we model the Universe to have values in 'n' different variables, then we say:

$$U^n = \{d_1, d_2, \ldots, d_n\}, \; |U^n| = n$$

We say that S^n is an arbitrary space which can be U^n or any other space we can think of. If a computation, 'M' has 'n' different steps, then it can be modelled as an n-dimensional space such that $S^n = M^n$ where each dimension, 'd_i' represents a step.

Not all spaces have non-zero values in every dimension. If a space has a zero-value in a particular dimension, it can still be modelled to have a parameter in that dimension with the value zero.

Whenever we consider variables which may or may not affect a space, we are thinking of the dimensions they may have values in. Any and every variable which takes a value can be a dimension or a space, as long as it has at least one parameter.

We can represent all variables which take real values on an axis. Even trivial variables which have the state of true or false can be modelled as dimensions. These are Boolean variables which can be represented on an axis by a '1' if the state of the variable is true or a '0' if the state is false.

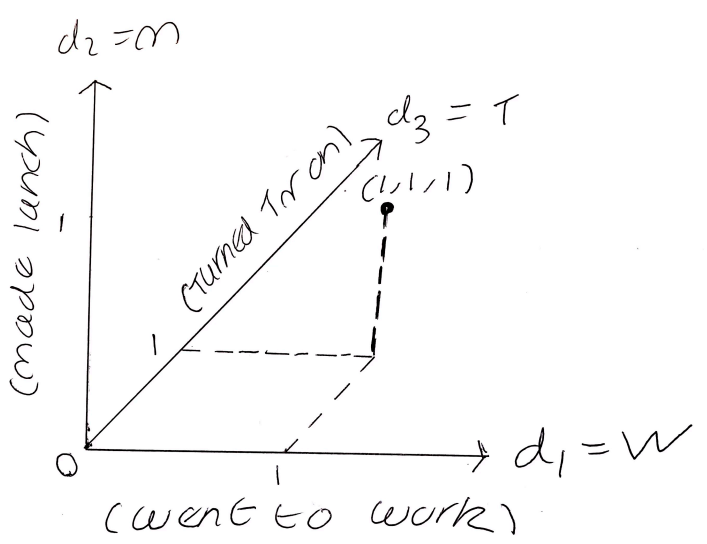
Examples of Boolean variables or dimensions include "Turning the TV on", "Going to work", "Making lunch", and so on. We can display the possible range of values for these types of variables or dimensions on the following page:

"Went to work" = $\boldsymbol{W} \in \{0, 1\}$

"Made lunch" = $\boldsymbol{M} \in \{0, 1\}$

"Turned the TV on" = $\boldsymbol{T} \in \{0, 1\}$

Suppose you went to work, made lunch and turned the TV on. Then $\boldsymbol{W} = 1$, $\boldsymbol{M} = 1$ and $\boldsymbol{T} = 1$. You could model these outcomes on a three-dimensional axes. Let $d_1 = \boldsymbol{W}$, $d_2 = \boldsymbol{M}$ and $d_3 = \boldsymbol{T}$. Then we have the following point on a graph:



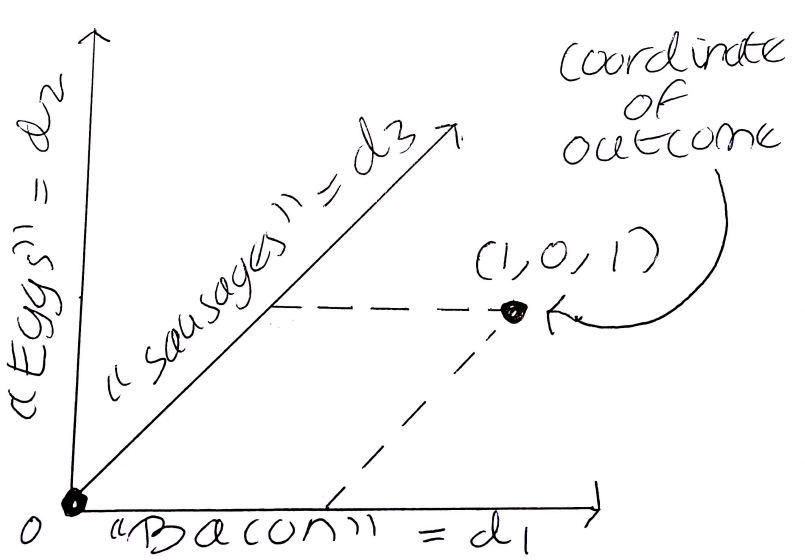
We notice that there is a three dimensional coordinate $(d_1, d_2, d_3) = (\boldsymbol{W}, \boldsymbol{M}, \boldsymbol{T}) = (1, 1, 1)$. We shall call this the 'coordinate of the outcome.'

Variables which take a set of string values can be represented numerically on axes too. Suppose the space "Breakfast" has three parameters "Bacon", "Eggs" and "Sausages", then the coordinates of the outcome will have '1's or '0's in the parameters of those which are true or false.

B^3 = Breakfast = {“Bacon”, “Eggs”, “Sausages”}

Let $\boldsymbol{M}$ = My breakfast, where $\boldsymbol{M}$ is a point on B^3

Let’s suppose the outcome of my breakfast includes bacon and sausages but no egg. Then the point $\boldsymbol{M}$ has the coordinates (1, 0, 1) on B^3:



Anything and everything that can manifest in the real world can be modelled as a dimension or a space.

A very large number of variables that we know of that apply to the real world take values which are real, Boolean or strings.

Even a trivial variable such as as “Turned the TV on” can be modelled as a space with a large number of dimensions; it will have dimensions in it’s own variables such as p*ressure applied to remote* or n*umber of people in the room as the TV was turned on* and any other variables we can think of.

Another way we can increase the number of parameters in a space is by considering variables in which the space does not have values; then we say that the space has zero-length dimensions in those parameters.

In the next chapter, we will explore ‘moments’; subset spaces of *n*-dimensional spaces which allow us to understand their parameters in greater detail.

2 — MOMENTS OF SPACES

ALL SPACES ARE SUBSET SPACES OF AN N-SPACE

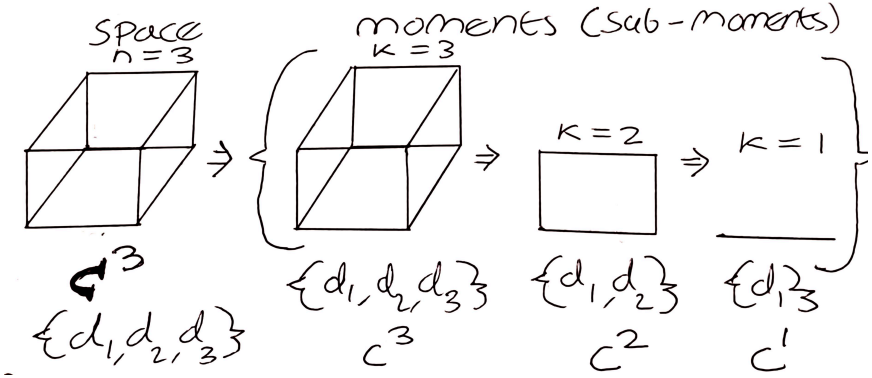
Consider a bowl of ice cream; not only are the values of its width, height and depth changing but so is its temperature and any other variable relating to it that we can think of. Let us denote the number of all the variables we can think of as 'n'.

Let us suppose we are only concerning ourselves with the width, height and depth of ice cream, but not all of the other variables, then we are analysing a three-dimensional subset of an n-dimensional bowl of ice cream. We shall refer to this subset as a 'moment' of '$\boldsymbol{k}$' dimensions, denoted 'm^k' where:

$$m^k = \{d_1, d_2, \ldots, d_k\},\ |m^k| = k,\ |m^k| \leq n$$
$$|m^k| \leq |S^n| = n$$
$$m^k \subseteq S^n,\ 1 \leq k \leq n$$

Moments exist in two varieties; sub-moments and parent moments. A 'sub-moment', 'm^k' is a subset space of a space 'S^n'. It has the same or fewer parameters as 'S^n', its parent moment.

Informally, a moment is what a space looked like before a dimension was added. A square, for instance, is a sub-moment of a cube — it has the first two dimensions included in the three-dimensional set of the cube; it is also a cube before depth is added:



We can state that a cube, C^3 has three sub-moments; a line, a square and itself. We can therefore derive the following subsets of C^3 which are sub-moments of the cube and their respective orders:

$$C^3 = \{d_1, d_2, d_3\},\ |C^3| = 3$$
$$c^3 = C^3 = \{d_1, d_2, d_3\},\ |c^3| = 3$$
$$c^2 = \{d_1, d_2\},\ |c^2| = 2$$
$$c^1 = \{d_1\},\ \ |c^1| = 1$$
$$c^1, c^2, c^3 \subseteq C^3$$

$$|c^k| \leq |C^3|,\ 1 \leq k \leq 3$$

The ice cream in three dimensions, before the dimensions of "time elapsed" and "temperature gained" were added, is a sub-moment of the melted ice cream space, 'I^5'.

Algorithms are very good examples of moments, since the sub-moments of the algorithm are simply the algorithm with the same or fewer steps, 'd_i', which will be explored in the fourth chapter *Algorithm of Outcomes*.

All n-dimensional spaces have n sub-moments. The melting ice cream space 'I^5' $= \{d_1, d_2, d_3, d_4, d_5\}$, from earlier, has the sub-moments $\{d_1\}$, $\{d_1, d_2\}$, $\{d_1, d_2, d_3\}$, $\{d_1, d_2, d_3, d_4\}$ and $\{d_1, d_2, d_3, d_4, d_5\}$.

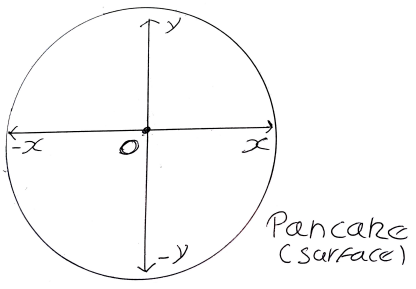
Sub-moments can be thought of as a series of steps where each successive sub-moment, a new parameter is added.

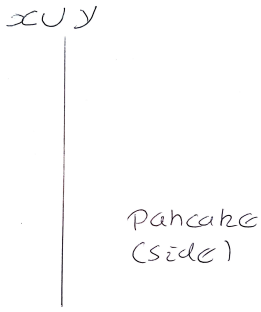
In the next chapter "Pancake Theory of Dimensions", we shall explore the concept of *unionising* dimensions.

3 — PANCAKE THEORY OF DIMENSIONS

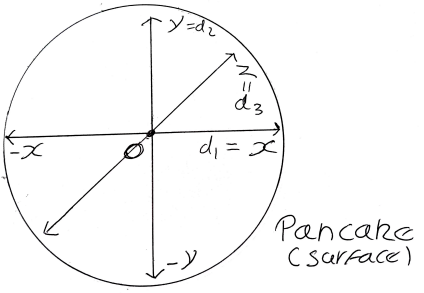
ALL DIMENSIONS CAN BE MODELLED AS FLAT

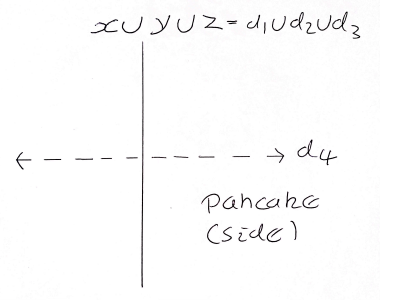
The Pancake Theory of Dimensions is such that all dimensions 'd_i', where $\boldsymbol{i} = \{1, 2, 3, \ldots\}$ can be modelled together as flat to appear as a one dimensional line. Consider a dimension as a pancake on its side. Then that pancake includes all the sub-dimensions of that dimension within its flat pancake surface while appearing as a line from its side. For instance, if the ***x,y*** axes are modelled as a pancake, then it includes all coordinates of ***x,y*** while appearing as a single line from the side of the pancake. We shall call this a unionisation of '***x***' and '***y***' or $\boldsymbol{x} \cup \boldsymbol{y}$*;* the left picture displays the surface of the pancake whereas the picture on the right displays its side in the ***z***-axis.



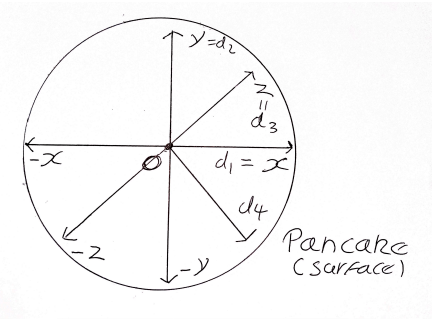


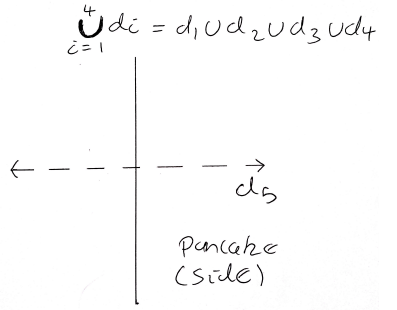
If the d_1,d_2,d_3 axes are modelled as a 3-dimensional pancake, then d_4 is a shift in the pancake $p_3 = d_1 \cup d_2 \cup d_3$ or $\cup_{i=1}^{3} d_i$



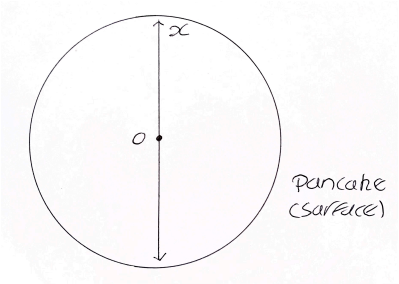


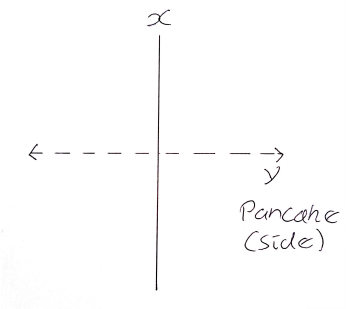
If the d_1,d_2,d_3,d_4 axes are modelled as a 4-dimensional pancake, then d_5 is a shift in the pancake $p_4 = d_1 \cup d_2 \cup d_3 \cup d_4$ or $\cup_{i=1}^{4} d_i$, and so on.



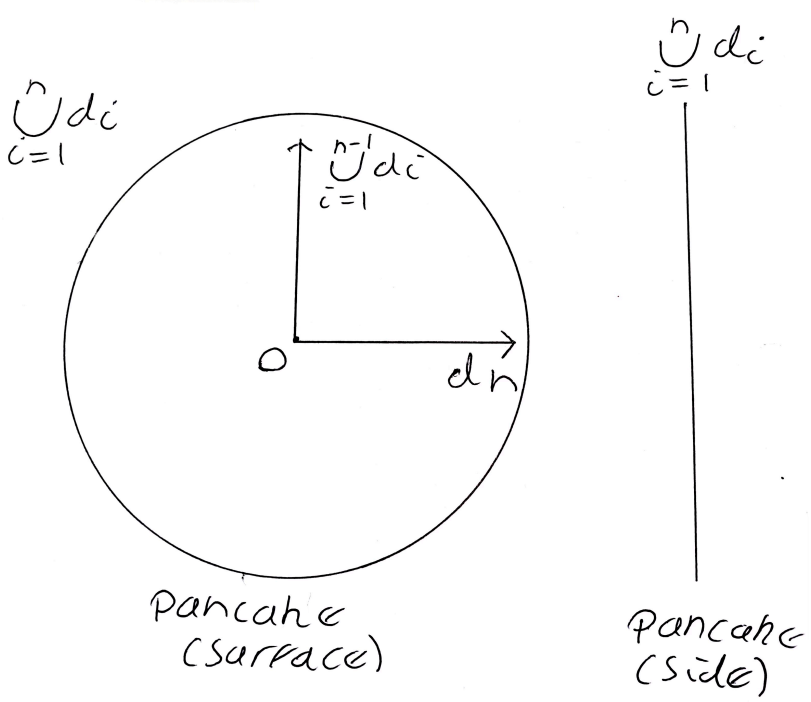


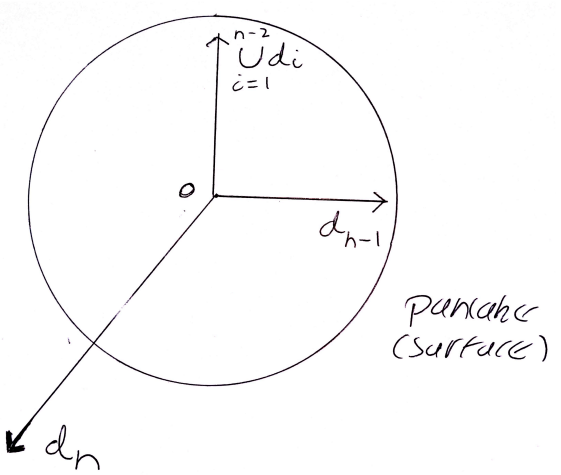
We shall also note that $\boldsymbol{x}$ is a 1-dimensional pancake in $\boldsymbol{y}$:



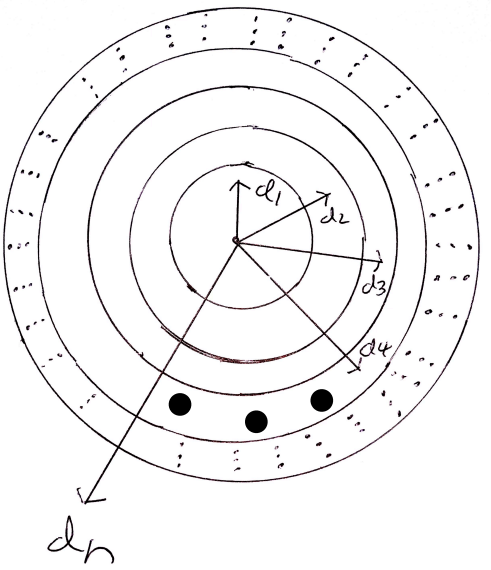


If the $d_1, d_2, \ldots, d_n$ axes is modelled as an n-dimensional pancake, then we have the pancake $p_n = d_1 \cup d_2 \cup \ldots \cup d_n$ or $\cup_{i=1}^{n} d_i$, where $n \to \infty$:





If we expand the pancakes out, then we have n pancakes, where '…' on the graph represents the pancakes between 'p_4' and 'p_{n-1}' and the dimensions between 'd_4' and 'd_n'.

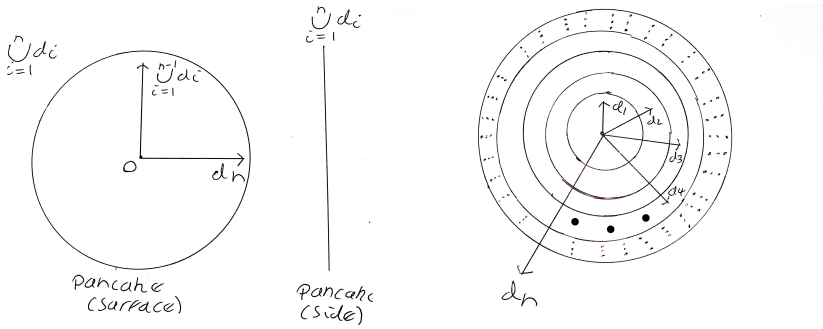


In everyday life we are constantly unionising dimensions as three-dimensional. When we are walking out of our house, one of the dimensions we are likely to be concerned with is the space between us and the outside of the house.
But there are other dimensions (variables) which we unionise with the dimension of space. For instance, the time that will pass in the moment, the temperature when we leave the house, even trivial variables such as the number of shoes we put on and the angle at which the door must be opened to walk through.

If we look close enough, the number of variables which we unionise with space are very large indeed.

As we pick apart these variables we 'split' the dimension of space into a larger number of variables (sub-dimensions) —- we shall call this technique 'deunionisation.'

'Deunionisation' is the reverse of 'unionisation' —- instead of modelling multiple dimensions as one, we split them into sub-dimensions. The two diagrams below illustrate both of these — the bottom middle picture is a complete unionisation of the dimensions of an n-dimensional space, whereas the bottom right picture is a deunionisation of an n-dimensional space. The leftmost picture is the surface view of the middle picture but with 'd_n' deunionised from '$\cup_{i=1}^{n-1} d_i$'.



Our lives mostly concern the space that surrounds us, so it is easy to unionise all dimensions, but humanly impossible to pick them all apart (i.e, completely deunionise).
At best we can partially deunionise the fully unionised dimension of space to the activities which concern us in the moments of our lives.

Some deunionisations, such as going about our daily activities, may be straightforward while others, such as being the best at a sport, becoming famous or other very specific desired outcomes for instance, may be far more complex as they may require knowledge of a higher number of variables than activities we perform which are more straightforward.

Irrespective of whether or not different outcomes can be objectively proven to be simple or complex relative to others, we can say that certain resulting outcomes of spaces will require knowledge of a larger number of parameters.

If we could know all variables and move in all parameters, no outcome would be complex and every *coordinate of outcome* would be simple to achieve. For instance, reading an encrypted letter would be no different to reading a letter written in plain English.

If the dimensions of a space 'S^n' or moment 'm^k 'are completely unionised then we say it has order '1.' If the dimensions of a space 'S^n' are completely deunionised, then we say that it has a maximum order which is a function of 'n.' We shall therefore derive the following minimums:

$$\min|m^k| = \min|S^n| = 1$$
$$|S^n| = \mathrm{n}$$

To demonstrate this, let us consider I^5, the melted ice cream space from Chapter 1.

Using the Pancake Theory, if we treat the 3D image of the melted ice cream as a flat pancake, then it can also be unionised as a single parameter such that:

$$I^5 = \{d_1, d_2, d_3, d_4, d_5\} = \{d_1 \cup d_2 \cup d_3, d_4, d_5\}$$

This reduces the order of the space 'I^5' to 3 and it's number of sub-moments to 3. We say that I^5 now has the sub-moments $\{d_1 \cup d_2 \cup d_3, d_4, d_5\}$, $\{d_1 \cup d_2 \cup d_3, d_4\}$ and $\{d_1 \cup d_2 \cup d_3\}$.

If we keep unionising the dimensions of 'I^5' then it will be reduced to an order of 1:

$$\begin{aligned} I^5 &= \{d_1 \cup d_2 \cup d_3, d_4, d_5\} \\ &= \{d_1 \cup d_2 \cup d_3 \cup d_4, d_5\} \\ &= \{d_1 \cup d_2 \cup d_3 \cup d_4 \cup d_5\} = \{p_5\} \end{aligned}$$

At this point the only sub-moment of I^5 is I^5 itself, so here we say that the order of its sub-moments equal the order of I^5:

$$|m^k| = |m^5| = |I^5|$$

If we unionise the dimensions of a space 'S^n', then we also unionise the dimensions of its moments. We say here that:

$$\begin{aligned} 1 \leq |I^5| &\leq \max |I^5| \\ 1 \leq |m^k| &\leq \max |m^k| \leq \max |I^5| \\ |m^k| &\leq |I^5| \end{aligned}$$

Whether we unionise the dimensions or not, it shall still be the case that:

$$1 \leq |S^n| \leq \max|S^n|$$
$$1 \leq |m^k| \leq \max|m^k| \leq \max|S^n|$$
$$|m^k| \leq |S^n|$$
$$|m^k| = |m^1|, |m^2|, \ldots, |m^n|$$
$$|m^n| = |S^n|$$
$$|m^1|, |m^2|, \ldots, |m^{n-1}| < |S^n|$$

We can unionise a space's dimensions in any order we wish, for instance, $I^5 = \{d_1 \cup d_3 \cup d_4, d_2, d_5\}$, which will have the moments $\{d_1 \cup d_3 \cup d_4, d_2, d_5\}$, $\{d_1 \cup d_3 \cup d_4, d_2\}$ and $\{d_1 \cup d_3 \cup d_4\}$.

Let's suppose we want to deunionise a space's dimensions. Then we just apply the same process but in reverse.

Suppose the ice cream, I^5, has only one, fully-unionised parameter 'd_1' and we wish to deunionise (split) the dimensions into two sub-dimensions, then:

$$I^5 = \{d_1\} = \{d_{1a} \cup d_{1b}\} = \{d_{1a}, d_{1b}\} = \{d_{1a}, d_{1b}\}$$
$$= \{D_1, D_2\}$$

Let's suppose we wish to deunionise I^5 until it has an element for each of its five observed dimensions. Then:

$$I^5 = \{D_1, D_2\} = \{D_{1a} \cup D_{1b}, D_{2a} \cup D_{2b}\}$$
$$= \{D_{1a}, D_{1b}, D_{2a}, D_{2b}\} = \{b_1, b_2, b_3, b_4\}$$
$$= \{b_{1a} \cup b_{1b}, b_2, b_3, b_4\} = \{b_{1a}, b_{1b}, b_2, b_3, b_4\}$$

$$= \{B_1, B_2, B_3, B_4, B_5\}$$

$$= I^5$$

Consider a dimension or variable such as “Turned TV on” from from “Chapter 1: *n*-spaces” takes a parameter ’t':

$$T = \{t\}$$

Then we can treat *T* as a one-dimensional space and deunionise its parameter ’*t*’ into four sub-dimensions; “Pressure applied to remote”, “Angle remote is held”, “Humans in the room” and
“Number of fingers holding remote” then:

Let P = “Pressure applied to remote”
Let A = “Angle remote is held”
Let H = “Humans in the room”
Let N = “Number of fingers holding remote”

Then $T = \{t\} = \{\text{P} \cup \text{A} \cup \text{H} \cup \text{N}\}$
$= \{\text{P}, \text{A}, \text{H}, \text{N}\}$

“Turning the TV on” now has four deunionised dimensions. We can continue deunionising further if we wish.
Since the number of variables we can think of as relating to the space are countless, we can see that identifying all the dimensions of the space ‘*T*’ would be impossible.

Complete unionisation of the space '*T*' is easy as we can just model the entire process of "Turning the TV on" as if it is one variable or dimension irrespective of the endless sub-variables or sub-dimensions in the process. The same applies for all other n-dimensional spaces or outcomes. We say that:

$$S^n = \{p_n\} = \{\cup_{i=1}^{n} d_i\} = \{d_1, d_2, \ldots, d_n\}$$
$$m^k = \{p_k\} = \{\cup_{i=1}^{k} d_i\} = \{d_1, d_2, \ldots, d_k\}$$
$$|\{p_k\}| \leq |\{p_n\}|$$

Using the inequalities from earlier we say that:

$$|m^k| = k, \; |S^n| = n$$

$$1 \leq |S^n| \leq n$$
$$1 \leq |m^k| \leq k \leq n$$

$$\max|m^k| \leq \max|S^n|$$

The order of a space is fluid, as long as the same dimensions are unionised or deunionised in its moments.
As we showed earlier, we can unionise all dimensions up to d_n on one axis, where n tends to infinity. The number of dimensions, '*n*', is finite in any given moment of a space, but tends to infinity.

If we decide to deunonise any dimension, 'd_i', of a space with *n* dimensions once, then the order will be '$n + 1$' and so we would treat '$n + 1$' as the new '*n*.'

Alternatively, if we decide to unionise any particular dimension 'd_i' of a space with n-dimensions once, then the order will become '$n - 1$', and we treat that as the new '*n*:'

Deunionisation:

$$S^n = \{d_1, d_2, d_3, \ldots, d_n\}$$
$$= \{d_{1a} \cup d_{1b}, d_2, d_3, \ldots, d_n\}$$
$$= \{d_{1a}, d_{1b}, d_2, d_3, \ldots, d_n\}$$
$$\text{Then } |S^n| = n + 1$$

Unionisation:

$$S^n = \{d_1, d_2, d_3, \ldots, d_n\}$$
$$= \{d_1 \cup d_2, d_3, \ldots, d_n\}$$
$$= \{p_2, d_3, \ldots, d_n\}$$
$$\text{Then } |S^n| = n - 1$$

If we deunionise any two dimensions d_i, once, then $|S^n| = n + 2$:

$$S^n = \{d_1, d_2, d_3, \ldots, d_n\}$$
$$= \{d_{1a} \cup d_{1b}, d_2, d_{3a} \cup d_{3b}, \ldots, d_n\}$$
$$= \{d_{1a}, d_{1b}, d_2, d_{3a}, d_{3b}, \ldots, d_n\}$$
$$\text{Then } |S^n| = n + 2$$

We say therefore that if we deunionise any 'ß' dimensions d_i, once, where ß $\in \{0, 1, 2, \ldots\}$, then:

$$|S^n| = n + ß$$

If we deunionise any two dimensions d_i, twice then

$|S^n| = n + 4$:

$$S^n = \{d_1, d_2, d_3, \ldots, d_n\}$$
$$= \{d_{1a} \cup d_{1b} \cup d_{1c}\ , d_{2a} \cup d_{2b} \cup d_{2c}\ , d_3, \ldots, d_n\}$$
$$= \{d_{1a}, d_{1b}, d_{1c}, d_{2a}, d_{2b}, d_{2c}\ , d_3, \ldots, d_n\}$$
$$\text{Then } |S^n| = n + 4$$

We can say therefore that if we deunionise any 'ß' dimensions d_i, twice, then:

$$|S^n| = n + 2ß$$

So far, we have experimented with deunionisation by adding dimensions, but what if we deunionise (split) all 'n' dimensions once? Then:

$$S^n = \{d_1, d_2, d_3, \ldots, d_n\}$$
$$= \{d_{1a} \cup d_{1b}\ , d_{2a} \cup d_{2b}, d_{3a} \cup d_{3b}, \ldots, d_{na} \cup d_{nb}\}$$
$$= \{d_{1a}, d_{1b}, d_{2a}, d_{2b}, d_{3a}, d_{3b}, \ldots, d_{na}, d_{nb}\}$$
$$\text{Then } |S^n| = 2n$$

Suppose we deunionise (split) all 'n' dimensions twice, then:

$$S^n = \{d_1, d_2, \ldots, d_n\}$$

$$= \{d_{1a} \cup d_{1b} \cup d_{1c}\ , d_{2a} \cup d_{2b} \cup d_{2c}\ , \ldots\ ,$$
$$d_{na} \cup d_{nb} \cup d_{nc}\}$$

$$= \{d_{1a}, d_{1b}, d_{1c}, d_{2a}, d_{2b}, d_{2c}, \ldots, d_{na}, d_{nb}, d_{nc}\}$$
$$\text{Then } |S^n| = 3n$$

We say, therefore that if we deunionise (split) all 'n' dimensions 'a' times, where a $\in \{0, 1, 2, \ldots\}$, then:

$$|S^n| = (a + 1)n$$

Therefore if we deunionise (split) all 'n' dimensions 'a' times and then deunionise any 'ß' dimensions d_i, once, then:

$$|S^n| = (a + 1)n + ß$$
$$a, ß \in \{0, 1, 2, \ldots\}$$

Or we could deunionise any 'ß' dimensions d_i, once and then deunionise (split) all individual dimensions 'a' times, in which case we would have:

$$|S^n| = (a + 1)(n + ß)$$

We say that 'a' can = 'n', as n is a natural number. Then if we deunionise (split) all 'n' individual dimensions 'n' times, we get the following result:

$$|S^n| = (a + 1)n = (n + 1)n = n^2 + n$$
$$\text{Therefore, } |S^n| = n^2 + n$$

Suppose we deunionise (split) all individual dimensions '0' times and then deunionise any dimension d_i once, then:

$$|S^n| = (0 + 1)n + 1$$

Suppose we deunionise (split) all 'n' dimensions 'n - 1' times and then unionise any dimension twice, it follows that:

$$|S^n| = (n - 1 + 1)n + 2$$
$$= n^2 + 2$$

If we deunionise all 'n' dimensions 'n^2 - 1' times and then unionise any dimension thrice then:

$$|S^n| = (n^2 - 1 + 1)n + 3 = n^3 + 3$$

If we deunionise all 'n' dimensions 'n^{k-1} - 1' times and then unionise any dimension k times then:

$$|S^n| = (n^{k-1} - 1 + 1)n + k$$
$$= n^k + k$$

We can therefore deduce that the maximum order of an n-dimensional space will be a polynomial in 'n', i.e, $O(n^p)$:

$$\max|S^n| = An^p + Bn^{p-1} + \ldots + K$$
$$\text{Where } A, B, \ldots, K \in \{0, 1, 2, \ldots\}$$
$$n \in \{1, 2, \ldots\}, p \in \{0, 1, 2, \ldots\}$$

Deunionisation is multiplicative or additive. Unionisation on other hand is the reverse —- a divisible or deductive process.

If we unionise a particular space's dimension d_i twice, then:

$$S^n = \{d_1, d_2, d_3, \ldots, d_n\}$$
$$= \{d_1 \cup d_2 \cup d_3 , d_4, \ldots, d_n\}$$
$$\text{Then } |S^n| = n - 2$$

If we unionise a particular dimension d_i 'ß' times then:

$$|S^n| = n + ß$$
$$ß \in \mathbf{Z}^- = \text{Set of negative Integers}$$

If we unionise each dimension d_i of a space S^n once, and n is even, then:

$$S^n = \{d_1, d_2, d_3, d_4, \ldots, d_n\}$$
$$= \{d_1 \cup d_2\ , d_3 \cup d_4, \ldots, d_{n-1} \cup d_n\}$$

$$\text{Therefore, } |S^n| = n/2$$

If n is odd, such that a dimension d_i is left over after pairing each of the dimensions, then we can just unionise it with one of the pairs. For example S^5:

$$S^5 = \{d_1, d_2, d_3, d_4, d_5\}$$
$$= \{d_1 \cup d_2\ , d_3 \cup d_4\ , d_5\}$$
$$= \{d_1 \cup d_2\ , d_3 \cup d_4 \cup d_5\}$$
$$\text{Therefore, } |S^5| = 2$$

Therefore, If we unionise each dimension d_i of a space S^n once and n is odd, then:

$$|S^n| = (n - 1)/2$$

If we keep unionising until all elements are unionised, then $|S^n| = 1$ whether n is even or odd, so the minimum is still 1 — therefore we can say that:

$$\min|S^n| = 1$$

Therefore,

$$1 \leq |S^n| \leq An^p + Bn^{p-1} + \ldots + K \in O(n^p)$$

Where $A, B, \ldots, K \in \{0, 1, 2, \ldots\}$
$p \in \{0, 1, 2, \ldots\}$
$n \in \{1, 2, \ldots\}$ = Set of natural numbers

Since the worst case running times of Turing machines in the context of the **P** versus **NP** problem are concerned with the maximum number of steps in a computation, and not the minimum, deunionisation is most relevant.

In summary:

$$\max|m^k| \leq \max|S^n| = An^p + Bn^{p-1} + \ldots + K$$

Where $A, B, \ldots, K \in \{0, 1, 2, \ldots\}$
$p \in \{0, 1, 2, \ldots\}$
$n \in \{1, 2, \ldots\}$ = Set of natural numbers

As the maximum is completely deunionised, $A, B, \ldots, K$ are non-negative.

Thus:

$$\max|m^k| \in O(k^p)$$

$$\max|S^n| \in O(n^p)$$

$$\min|m^k| = \min|S^n| = 1$$

$$1 \leq |S^n| \leq \max|S^n| \in O(n^p)$$

$$1 \leq |m^k| \leq \max|m^k| \leq \max|S^n| \in O(n^p)$$

and, as was seen previously:

$$p_n = d_1 \cup d_2 \cup \ldots \cup d_n \text{ or } \cup_{i=1}^{n} d_i, \text{ where n} \rightarrow \infty$$

$$S^n = \{p_n\} = \{\cup_{i=1}^{n} d_i\} = \{d_1, d_2, \ldots, d_n\}$$
$$m^k = \{p_k\} = \{\cup_{i=1}^{k} d_i\} = \{d_1, d_2, \ldots, d_k\}$$
$$|\{p_k\}| \leq |\{p_n\}|$$

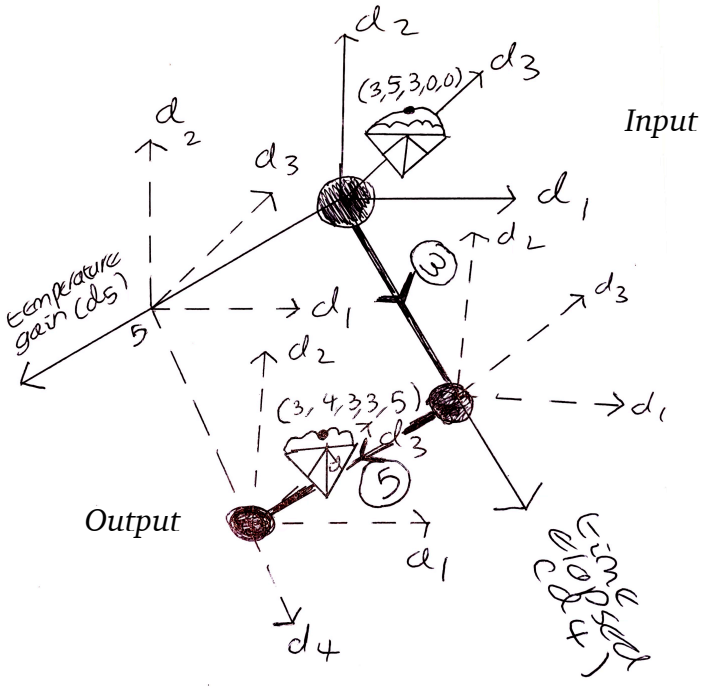
In the next chapter we will cover the General Algorithm of Outcomes.

4 — ALGORITHM OF OUTCOMES

IF A SPACE IS N-DIMENSIONAL THEN IT HAS A PATH OF N PARAMETERS

All n-dimensional spaces have a path of n parameters, where every parameter can be modelled as a step. The origin point shall be modelled as the starting node (which contains the input). The end node contains the output.

The *5-dimensional melting ice cream space* from the first chapter has a 3-node path $\{d_1 \cup d_2 \cup d_3, d_4, d_5\}$:



We say that the inputs are d_1, d_2, d_3 in the starting node and the *3-dimensional ice cream space* is a unionised input 'D_1' of the three sub-inputs d_1, d_2, d_3

Thus: $\{d_1 \cup d_2 \cup d_3, d_4, d_5\} = \{D_1, D_2, D_3\}$

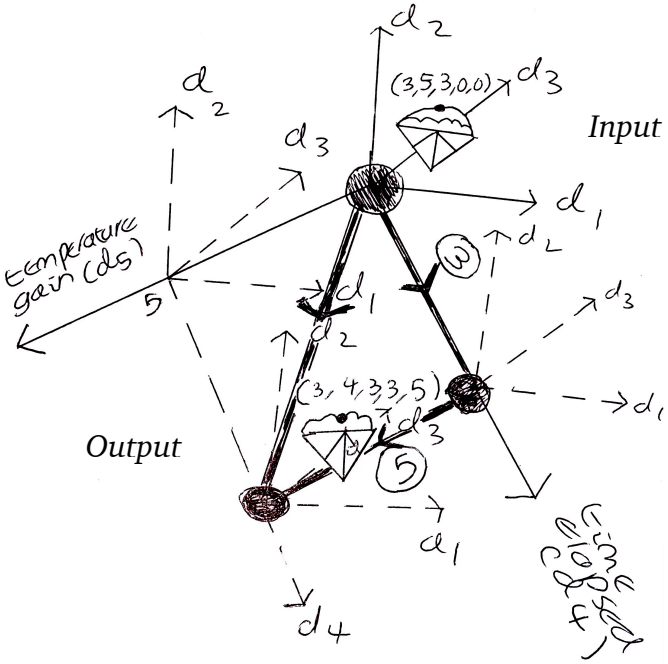
And, therefore, the steps referred to are as follows:

Step 1 (Input): Image of the ice cream $D_1 = d_1 \cup d_2 \cup d_3$,
Where d_1, d_2, d_3 are inputs and D_1 is the parent input.

Step 2: Time elapses by 3 units:
$(d_1 \cup d_2 \cup d_3, d_4 = 3) = (D_1, D_2)$

Step 3 (Output): Temperature goes up by 5 units:
$(d_1 \cup d_2 \cup d_3, d_4 = 3, d_5 = 5) = (D_1, D_2, D_3)$

We create a path with one less arc by unionising d_4 and d_5:



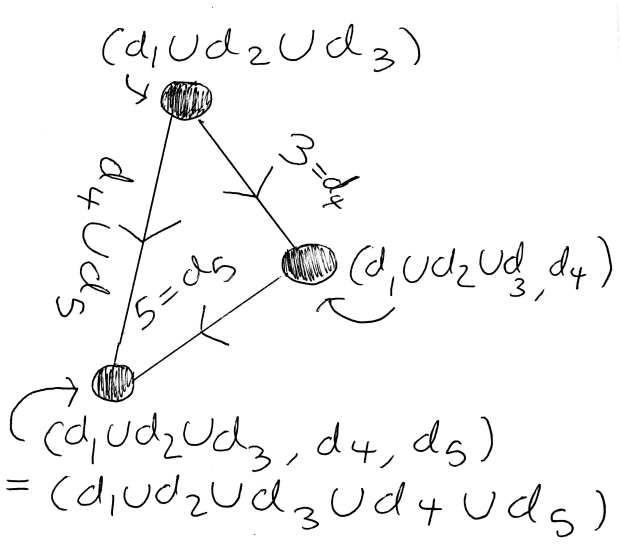
The *3-step Ice Cream Melting Algorithm* is now a *2-step algorithm:*

Step A = Step 1 **(Input)**: Image Of The Ice Cream:
$C_1 = d_1 \cup d_2 \cup d_3$, where d_1, d_2, d_3 are inputs and C_1 is the parent input.

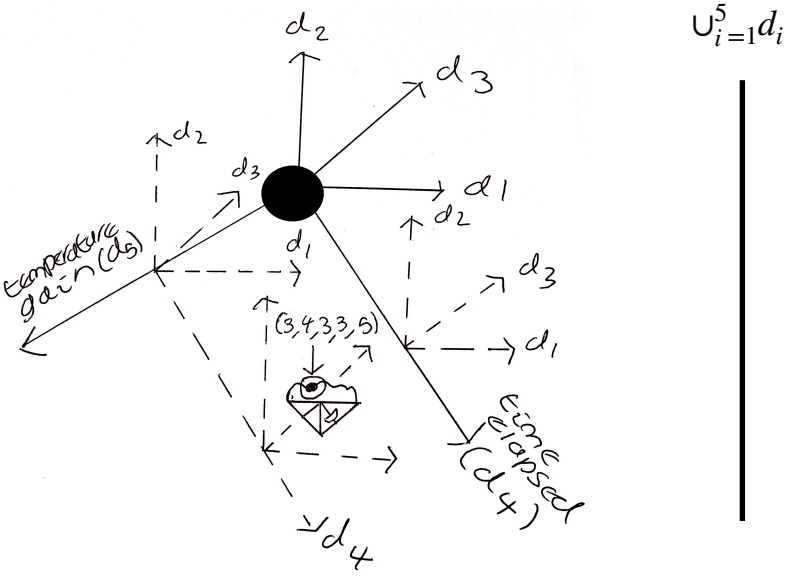
Step B = Step 2 $\cup$ 3 **(Output)**: Melted Ice Cream:
Time elapses by 3 units, Temperature goes up by 5 units:
$(d_1 \cup d_2 \cup d_3, d_4 \cup d_5) = (C_1, C_2)$

The coordinate of outcome for this path is
$(d_1 \cup d_2 \cup d_3\ ,\ d_4 \cup d_5)$

We can summarise the two paths, we have created below. We notice that, at the output node, we can unionise from the original *3-step path* we proposed to a *1-step path*.



If we unionise '$d_1 \cup d_2 \cup d_3$' with '$d_4 \cup d_5$' then the 1-step path with coordinate of outcome $(d_1 \cup d_2 \cup d_3 \cup d_4 \cup d_5)$ is a node at the origin point, with all five dimensions extending out of it and the image of the melted ice cream. In a space with fully-unionised dimensions, the input and the output are the same. The image on the bottom left is the deunionised form. In the unionised form, we simply think of it as the side of a pancake (a line) with the image on the surface.



As seen, the *2-step Ice Cream Melting Algorithm* is now a *1-step algorithm:*

*1-**Step Algorithm*** = Step 1 ∪ 2 ∪ 3 **(Input and Output)**: Image of the melted ice cream with coordinate of outcome $(d_1 \cup d_2 \cup d_3 \cup d_4 \cup d_5)$

In the *Algorithm Of Outcomes* the number of parameters are the number of steps, including zero-containing parameters. A space with its dimensions completely unionised such that it has one fully-unionised dimension, will have a path with just one step which is both the input and the output. A space with '*N*' unionised dimensions will have a path with '*N*' parameters and therefore '*N*' steps.

Equally, we can say that such a space has '*N*' deunionised dimensions, which still will have a path with '*N*' steps.

As we established in the previous chapter, a space with fully-deunionised dimensions will have a path with a number of steps in a polynomial of '*N*', which are the maximum number of steps for that path.

An *Algorithm Of Outcome* graph is an *N*-dimensional space 'M^N' with a path of '*N*' steps such that:

$$M^N = \{D_1, D_2, \ldots, D_N\},\ |M^N| = N$$

$$\max|M^N| = A(N^P) + B(N^{P-1}) + \ldots + K$$

We say that D_1 is the input such that:

$$D_1 = \{d_1 \cup d_2 \cup \ldots \cup d_n\} = \{d_1, d_2, \ldots, d_n\}$$
$$t_M(D_1) = |M^N| = N = \text{Number of steps for computation to halt.}$$

Where '$d_1 \cup d_2 \cup \ldots \cup d_n$' is the unionised input and '$d_1, d_2, \ldots, d_n$' are the deunionised inputs:

$$M^N = \{D_1, D_2, \ldots, D_N\}$$
$$= \{d_1 \cup d_2 \cup \ldots \cup d_n, D_2, D_3, \ldots, D_N\}$$
$$= \{d_1, d_2, \ldots, d_n, D_2, D_3, \ldots, D_N\}$$

Then we say that:

$$\max|M^N| = \max|\{d_1, d_2, \ldots, d_n, D_2, D_3, \ldots, D_N\}|$$
$$= \max|\{d_1, d_2, \ldots, d_n\}| + \max|\{D_2, D_3, \ldots, D_N\}|$$

$$= A_1(n^p) + B_1(n^{p-1}) + \ldots + K_1$$
$$+ A_2(N-1)^P + B_2(N-1)^{P-1} + \ldots + K_2$$

$$= \max|M^N| = A_1(n^p) + B_1(n^{p-1}) + \ldots + ß$$
$$\text{where } ß = K_1 + A_2(N-1)^P + B_2(N-1)^{P-1} + \ldots + K_2$$

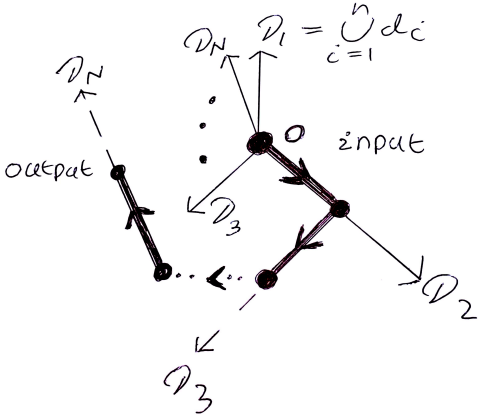
Or

$$\max|M^N| = A_2(N-1)^P + B_2(N-1)^{P-1}) + \ldots + \mu$$
$$\text{where } \mu = K_2 + A_1(n^p) + B_1(n^{p-1}) + \ldots + K_1$$

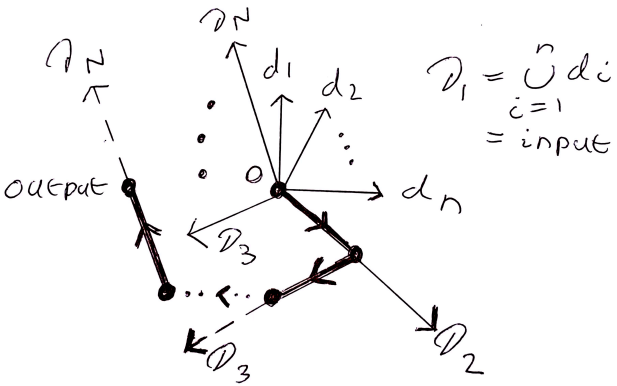
$$ß, A_1, B_1, \ldots, K_1 \in \{0, 1, 2, \ldots\}$$
$$\mu, A_2, B_2, \ldots, K_2 \in \{0, 1, 2, \ldots\}$$
$$p, P \in \{0, 1, 2, \ldots\}$$
$$n, N \in \{1, 2, \ldots\}$$
$$ß, \mu \in \{0, 1, 2, \ldots\}$$

We note that the *Algorithm Of Outcomes* is a polynomial in its steps '*N*' and inputs '*n*'.

The *Algorithm Of Outcomes* can be graphically represented as follows:



The graph above represents the steps of M^N (i.e, $D_1, D_2, \ldots, D_N$). The graph below represents M^N but with the sub-inputs of 'D_1' deunionised. The vertical dots on the graph below represent the inputs in between d_2 and d_n on the right and the steps in between D_3 and D_N on the left. The dotted arrow (..<..) represents the arcs in between.



We have the input node at the origin point which the unionised input 'D_1' and all other 'D_i' extend out of.

In both diagrams we have a N-step path, with the final arc parallel to the D_N axis, joining to the output node.

Suppose we are given the task of graphically representing a program using the *Algorithm of Outcomes*.

The following program, written in 'Swift', calculates the volume of a tesseract, based on four different parameters —- width, length, depth and time:

```
struct Tesseract {

    var width: Double
    var length: Double
    var depth: Double
    var time: Double

    func volumeTesseract() -> Double {
        return width * length * depth * time
    }

}

let volTesseract = Tesseract(width: 5.0, length: 5.0, depth:
5.0, time: 5.0)

let volumeTesseract = volTesseract.volumeTesseract()

var volTesseractString: String = "The volume of the tesseract
is \(volumeTesseract) units ^ 4"

print(volTesseractString)
```

We can start by denoting the steps 'D_i':

$D_1 = \{d_1, d_2, d_3, d_4\}$ = {width, length, depth, time},

D_2 = volumeTesseract(), D_3 = volTesseract,

D_4 = volumeTesseract, D_5 = volTesseractString,

D_6 = print(volTesseractString)

Next, we write out the following steps:

Step 1 (Input): $D_1 = d_1 \cup d_2 \cup d_3 \cup d_4$,
Where $\{d_1, d_2, d_3, d_4\}$ = {width, length, depth, time}
are inputs and D_1 is the parent input.

Step 2: D_2 = volumeTesseract()
($d_1 \cup d_2 \cup d_3 \cup d_4$, volumeTesseract()) = ($D_1$, D_2)

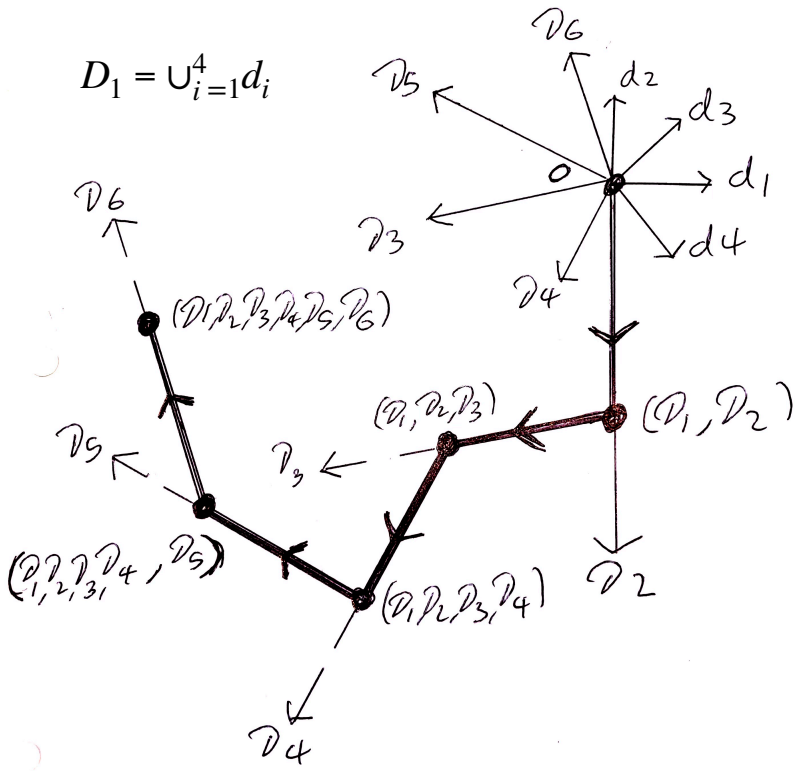
Step 3: D_3 = volTesseract
($d_1 \cup d_2 \cup d_3 \cup d_4$, volumeTesseract(), volTesseract)
= (D_1, D_2, D_3)

Step 4: D_4 = volumeTesseract
($d_1 \cup d_2 \cup d_3 \cup d_4$, volumeTesseract(), volTesseract,
volumeTesseract)
= (D_1, D_2, D_3, D_4)

Step 5: D_5 = volTesseractString
($d_1 \cup d_2 \cup d_3 \cup d_4$, volumeTesseract(), volTesseract,
volumeTesseract, volTesseractString)
= (D_1, D_2, D_3, D_4, D_5)

Step 6 (Output): D_6 = print(volTesseractString)
($d_1 \cup d_2 \cup d_3 \cup d_4$, volumeTesseract(), volTesseract,
volumeTesseract, volTesseractString,
print(volTesseractString))
= (D_1, D_2, D_3, D_4, D_5, D_6)

Finally, we show the *6-step algorithm* graphically:



We can reduce the steps of this algorithm in the same way as the *Ice Cream Melting Algorithm,* i.e, unionising until we have a *1-step algorithm* with *coordinate of outcome* $(D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5 \cup D_6)$, and all other possible permutations of unionisations we can think of.

We can show that the maximum number of steps of the program 'D_i' are a polynomial of the input.

We say that D_1 is the input such that:

$$D_1 = \{d_1 \cup d_2 \cup d_3 \cup d_4\} = \{d_1, d_2, d_3, d_4\},\ |D_1| = 4$$

We say that the computation 'M' has six steps such that:

$$M^6 = \{D_1, D_2, D_3, D_4, D_5, D_6\}$$

$$t_M(D_1) = \text{number of steps for '}M^6\text{' to halt} = |M^6| = 6$$

$$M^N = \{D_1, D_2, D_3, D_4, D_5, D_6\}$$
$$= \{d_1 \cup d_2 \cup d_3 \cup d_4, D_2, D_3, D_4, D_5, D_6\}$$
$$= \{d_1, d_2, d_3, d_4, D_2, D_3, D_4, D_5, D_6\}$$

We say the maximum number of inputs is $\max|D_1|$, where:

$$\max|D_1| = A_1(4^p) + B_1(4^{p-1}) + \ldots + K_1$$
$$|\{D_2, D_3, D_4, D_5, D_6\}| = 5$$

$$\max|\{D_2, D_3, D_4, D_5, D_6\}| = A_2(5^P) + B_2(5^{P-1}) + \ldots + K_2$$

Therefore,

$$\max|M^N| = \max|\{d_1, d_2, d_3, d_4, D_2, D_3, D_4, D_5, D_6\}|$$

$$= \max|\{d_1, d_2, d_3, d_4\}| + \max|\{D_2, D_3, D_4, D_5, D_6\}|$$

$$= A_1(4^p) + B_1(4^{p-1}) + \ldots + K_1$$
$$+ A_2(5^P) + B_2(5^{P-1}) + \ldots + K_2$$

$$\text{Let } ß = K_1 + A_2(5^P) + B_2(5^{P-1}) + \ldots + K_2$$

$$\text{Then } \max|M^N| = A_1(4^p) + B_1(4^{p-1}) + \ldots + ß$$
$$= T_M(n = 4)$$

We see therefore that the algorithm for this program runs in Polynomial Time. In the following chapter, using all of my findings in this paper so far, I will show that all **NP** problems belong to the **P** class of problems.

5 — P = NP

Let '$\sum$' be an alphabet. An alphabet is a set of elements such as {0, 1}.

Then let 'd_i' be an element of an alphabet '$\sum$' such that:

$$d_i \in \sum$$

$$\text{Then } d_1, d_2, \ldots, d_n \in \sum$$

If $\sum$ = {0, 1} for instance, then one combination could be:

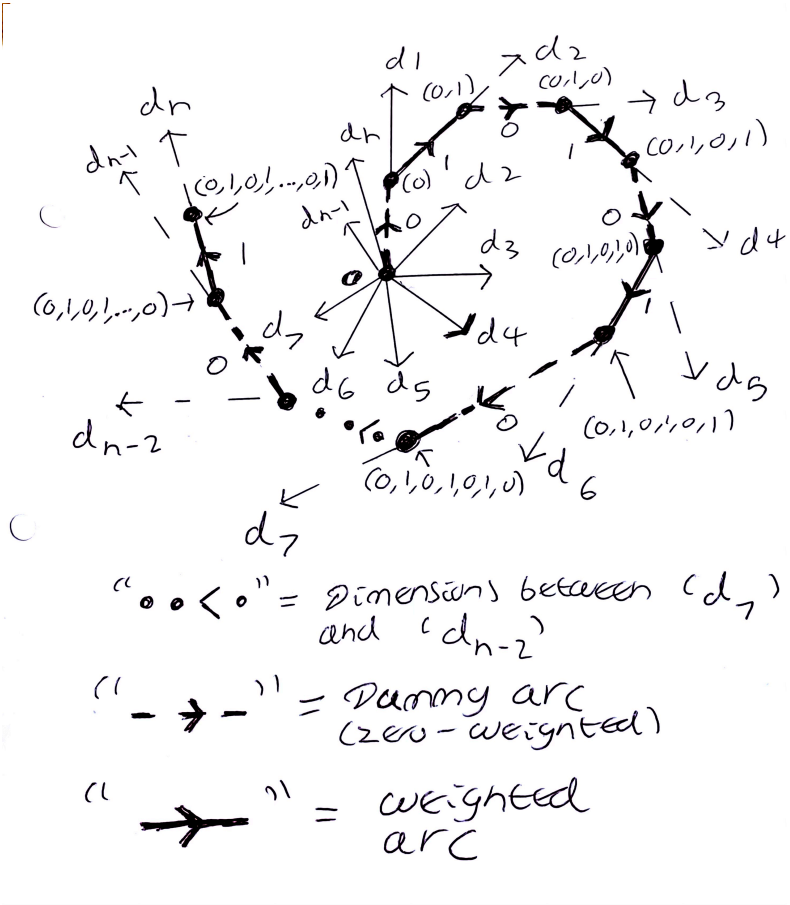
$$0, 1, 0, 1, 0, 1, \ldots., 0, 1 \in \sum$$

$$0, 1, 0, 1, 0, 1, \ldots., 0, 1$$
$$= d_1, d_2, d_3, d_4, d_5, d_6, \ldots, d_{n-1}, d_n \in \sum = \{0, 1\}$$

Where 'n' represents the size of the combination and '$d_1, d_2, \ldots, d_n$' is a possible combination of elements of an alphabet '$\sum$'

As a result of this example, we get a space with the coordinate of outcome
(0, 1, 0, 1, 0, 1, …., 0, 1), as shown graphically on the next page.

We use dummy arcs (dashed lines) to represent the zero-weighted parameters.



Let 'v' be a string input of length 'n' where 'v' = $d_1 d_2 \ldots d_n$ and $d_i \in \sum$

Then the string 'v' corresponding with the coordinate of outcome (0, 1, 0, 1, 0, 1, …., 0, 1) is
'v' = 010101…01

Therefore, using the *Pancake Theory of Dimensions*, we model 'v' as a unionised input of sub-inputs $d_1, d_2, \ldots, d_n$ such that 'v' = {0, 1, 0, 1, 0, 1, …., 0, 1}
= {0 ∪ 1 ∪ 0 ∪ 1 ∪ 0 ∪ 1 ∪ … ∪ 0 ∪ 1}

Then we have the following fully-unionised graph of 'v':

$$v = 0 \cup 1 \cup 0 \cup 1 \cup \ldots \cup 0 \cup 1$$

We recognise the origin point as the starting node which contains the input string 'v', which is a complete unionisation of it's sub-inputs {0, 1, 0, 1, 0, 1, …., 0, 1}.

We say that 'v' is an input of a computation 'M^N', which we describe as a space of 'N' steps $\{D_1, D_2, \ldots, D_N\}$, where each D_i is a step.

$$M^N = \{D_1, D_2, \ldots, D_N\},$$
$$t_M(v) = |M^N| = N$$
$$T_M(n) = \max\{t_M(w) \mid w \in \sum^n\}$$

By the *Pancake Theory of Dimensions*, the maximum order of a space 'S^N' is a complete deunionisation of it's dimensions, i.e, a polynomial in 'N':

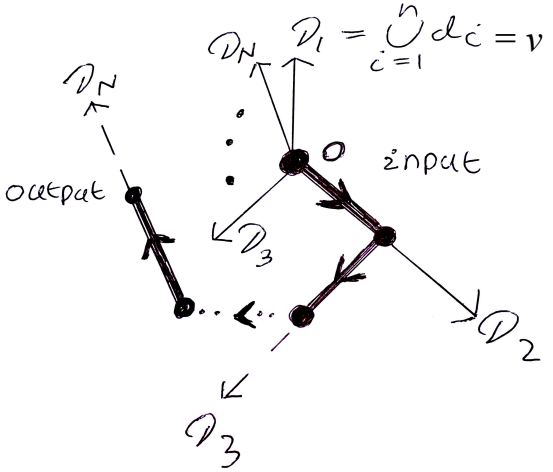
$$\max|S^N| = A(N^P) + B(N^{P-1}) + \ldots + K$$

Where A, B, …, K $\in \{0, 1, 2, \ldots\}$
$N \in \mathbf{Z}^+$ = Set of natural numbers
$p \in \{0, 1, 2, \ldots\}$

Therefore, for a space $S^N = M^N$:

$$\max|M^N| = A(N^P) + B(N^{P-1}) + \ldots + K = T_M(n = |v|)$$

We can see that v as a fully-unionised parameter is a parameter 'D_1' of the computation 'M^N':



Such that $v = D_1 \in M^N$

$$\text{Therefore, } M^N = \{D_1, D_2, D_3, \ldots, D_N\}$$
$$= \{v, D_2, D_3, \ldots, D_N\}$$
$$= \{\{0 \cup 1 \cup 0 \cup 1 \cup 0 \cup 1 \cup \ldots \cup 0 \cup 1\}, D_2, D_3, \ldots, D_N\}$$
$$= \{\{0, 1, 0, 1, 0, 1, \ldots., 0, 1\}, D_2, D_3, \ldots, D_N\}$$
$$= \{0, 1, 0, 1, 0, 1, \ldots., 0, 1, D_2, D_3, \ldots, D_N\}$$

We then can unionise the elements D_i which are not 'v'

$$= \{0, 1, 0, 1, 0, 1, \ldots., 0, 1, D_2 \cup D_3 \cup \ldots \cup D_N\}$$
$$= \{0, 1, 0, 1, 0, 1, \ldots., 0, 1, \cup_{i=2}^{N} D_i\},$$
which has order $n + 1$ in terms of 'n.'

$$\{0, 1, 0, 1, 0, 1, \ldots., 0, 1, \cup_{i=2}^{N} D_i\}$$
$$= \{0, 1, 0, 1, 0, 1, \ldots 0, 1, D_2, \ \cup_{i=3}^{N} D_i\},$$
which has order $n + 2$ in terms of 'n'.

Therefore, $\{0, 1, 0, 1, 0, 1, \ldots., 0, 1, D_2, D_3, \ldots, D_N\}$
has order $n + N$ - 1 in terms of 'n'.

We see that without D_1, $|\{D_2, D_3, \ldots, D_N\}| = N$ - 1

Therefore, $\max|\{D_2, D_3, \ldots, D_N\}|$
$$= A(N-1)^p + B(N-1)^{p-1} + \ldots + K$$

$$\max|M^N|$$
$$= \max|\{0, 1, 0, 1, 0, 1, \ldots., 0, 1, D_2, D_3, \ldots, D_N\}|$$
$$= \max|\{0, 1, 0, 1, 0, 1, \ldots., 0, 1\}|$$
$$+ \max|\{D_2, D_3, \ldots, D_N\}|$$

$$= A_1 n^p + B_1 n^{p-1} + \ldots + K_1$$
$$+ A_2(N-1)^P + B_2(N-1)^{P-1} + \ldots + K_2$$

$$= A_1 n^p + B_1 n^{p-1} + \ldots$$
$$+ (K_1 + A_2(N-1)^P + B_2(N-1)^{P-1} + \ldots + K_2)$$

Let $ß = K_1 + A_2(N-1)^P + B_2(N-1)^{P-1} + \ldots + K_2$

$$A_1, B_1, \ldots, K_1 \in \{0, 1, 2, \ldots\}$$
$$ß, A_2, B_2, \ldots, K_2 \in \{0, 1, 2, \ldots\}$$
$$N \in \mathbf{Z}^+ = \text{Set of natural numbers}$$
$$p, P \in \{0, 1, 2, \ldots\}$$

Therefore,

$$\max|M^N| = A_1 n^p + B_1 n^{p-1} + \ldots + ß$$

Let $a, b, \ldots, k = A_1, B_1, \ldots, ß$

$$A_1 n^p + B_1 n^{p-1} + \ldots + ß$$
$$= an^p + bn^{p-1} + \ldots + k$$

Therefore,

$$T_M(n = |v|) = an^p + bn^{p-1} + \ldots + k$$

$$p, a, b, \ldots, k \in \{0, 1, 2, \ldots\}$$
$$n \in \{1, 2, \ldots\} = \text{Set of natural numbers}$$

Therefore, $T_M(n = |v|) \in O(n^p)$

The string 'w' $= d_1 d_2 \ldots d_n$ has the corresponding *Coordinate of Outcome* $(d_1, d_2, \ldots, d_n)$, so we do exactly the same as we did with 'v.'

By the *Pancake Theory of Dimensions*, $\{d_1, d_2, \ldots, d_n\} = \{d_1 \cup d_2 \cup \ldots \cup d_n\}$, therefore the unionised input 'w' is the first step 'D_1' in a computation 'M^N' $= \{D_1, D_2, \ldots, D_N\}$
$= \{w, D_2, D_3, \ldots, D_N\} = \{d_1, d_2, \ldots, d_n, D_2, D_3, \ldots, D_N\}$
where $|\{d_1, d_2, \ldots, d_n, D_2, D_3, \ldots, D_N\}| = n + N - 1$

Therefore, $\max|M^N| = \max(t_M(w))$

$$= \max |\{d_1, d_2, \ldots, d_n, D_2, D_3, \ldots, D_N\}|$$

$$= \max |\{d_1, d_2, \ldots, d_n\}| + \max |\{D_2, D_3, \ldots, D_N\}|$$

$$= A_1 n^p + B_1 n^{p-1} + \ldots + K_1$$
$$+ A_2(N-1)^P + B_2(N-1)^{P-1} + \ldots + K_2$$

$$ß = K_1 + A_2(N-1)^P + B_2(N-1)^{P-1} + \ldots + K_2$$

Therefore,

$$\max|M^N| = A_1 n^p + B_1 n^{p-1} + \ldots + ß$$

Let $a, b, \ldots, k = A_1, B_1, \ldots, ß$

Therefore,

$$T_M(n = |w|) = \max|M^N| = an^p + bn^{p-1} + \ldots + k$$

$p, a, b, \ldots, ß \in \{0, 1, 2, \ldots\}$

$n \in \{1, 2, \ldots\}$ = Set of natural numbers

$$T_M(n) \in O(n^p)$$

Therefore, for every string input 'w' of $d_1, d_2, \ldots, d_n$, where $d_1, d_2, \ldots, d_n$ are elements of an alphabet '$\sum$' :

$$T_M(n = |w|) \in O(n^p)$$

Therefore, there exists a language $L = L(M)$ for all computations 'M' which accept a string input 'w', therefore all problems are in class '**P**'.

Therefore,

P = **NP** in all cases.

CONCLUSION

In summary, all spaces can be modelled to have parameters in '*n*' variables or dimensions, and anything we can think of can be modelled as a space with a natural number of dimensions (parameters). We refer to these as *n*-dimensional spaces or *n*-spaces. All *n*-dimensional spaces can be modelled on *n*-dimensional axes with '*n*' dimensions, where each dimension d_i represents each parameter of the space.

By the *Pancake Theory of Dimensions*, dimensions (parameters) can be deunionised (split) into sub-dimensions (sub-parameters) or unionised into one parent dimension (parameter).

As deunionisation is a multiplicative or additive process, increasing the number of parameters in an *n*-dimensional space solely through deunionisation always results in a polynomial in '*n*' of non-negative order; so the maximum order of the set of parameters in an *n*-dimensional space must be a polynomial in '*n*'. Unionisation on the other hand, is a deductive or divisive process that, when continually performed on parameters in an *n*-dimensional space, results in a one-dimensional space.

A space with inputs as parameters, such as a computer program, or of any other scenario we can think of which involves inputs, can therefore have its inputs deunionised (split) into sub-inputs or unionised into one input.

An input string '*w*' of length '*n*', therefore, can be modelled as a unionised input (parent input) of '*n*' sub-inputs where each sub-input 'd_i' is an element of an alphabet '$\sum$.'

We say that 'w' is an input of a computation 'M', which has 'N' parameters D_i, which we denote as 'M^N', where each D_i is a step.

We say that all n-dimensional spaces comprise a path called an *Algorithm of Outcome* that travels across all 'n' parameters (dimensions) concerning the space.

An N-dimensional space, 'M^N' has an *Algorithm of Outcome* that travels across all of its '$\boldsymbol{N}$' parameters (dimensions).

An *Algorithm of Outcome* for a space 'S^N', such as 'M^N', will have an input node from which a unionised parent input 'D_1' of 'n' sub-inputs extends from the node. We say that the first step in the *Algorithm of Outcome* is the unionised input 'D_1' (in the context of string inputs, '$D_1 = w$').

Then the steps '$D_2, D_3, \ldots, D_N$' represent the input '$D_1 = w$' under the influence of each successive dimension 'D_i'. As $t_M(w)$, the number of steps, 'N' for the computation M^N to halt, can be increased or multiplied by a scalar through deunionising (splitting) the steps 'D_i' into sub-steps. We say, therefore, that the order of a computation M^N is always a polynomial in 'N'; so the order of 'M^N' is an element of $O(N^P)$, where p is a whole number greater than or equal to zero. Therefore, the maximum number of steps in a computation M^N, must also be a polynomial in 'N', which is also an element of $O(N^P)$.

As the number of inputs in 'D_1' (which are 'n'), can be increased or multiplied by a scalar (such as 'n' itself) by deunionising (splitting) the inputs 'd_i' into sub-inputs, we can say that the order of the input D_1 is always a polynomial in 'n', and so the order of 'D_1' is an element of $O(n^p)$.

Therefore, the maximum number of sub-inputs in an input D_1, must also be a polynomial in 'n', which is also an element of $O(n^p)$. As the unionised input 'D_1' is an element of 'M^N', we showed that $T_M(n)$, the worst case running time or maximum order of 'M^N', can be written in terms of 'n'. This proves that $T_M(n)$, the maximum number of steps in any computation 'M^N', is a polynomial in terms of 'n' for any input string 'w' of length 'n'. *Therefore,* **P** = **NP**.

ABOUT THE AUTHOR

India Soale is a BSc Mathematics student at the University of Sussex and iOS App Developer.

In 2019, as part of the first year of her degree, she completed a project on the Fourth Dimension, covering four-dimensional spaces and calculating the volumes of their spaces.

In the Summer of 2020 she became very interested in the Riemann Hypothesis and spent a great deal of time experimenting with different infinite sums.

In January of 2021, during the Coronavirus Pandemic lockdowns she began experimenting with L-functions. At the same time she also developed an interest in the P Versus NP problem.

She postponed her efforts to solve the Riemann Hypothesis, in order to concentrate on her Mathematics degree and the development of her iOS App "Peekner."

Since then, she has used her holidays to resume her attempts to solve the Riemann Hypothesis.

During the Summer of 2022, after much thought about the true nature of dimensions, variables and parameters in relation to a space, she became interested in the P Versus NP Problem, and worked on devising an algorithm that would implement all the dimensions of a space. Her subsequent work on this problem has resulted in this paper.

BIBLIOGRAPHY

[Stephen Cook - “Statement of the Problem”]:
http://www.claymath.org/sites/default/files/pvsnp.pdf